Journal of Theoretical and Applied Information Technology
31st May 2018. Vol.96. No 10
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A RULE SELECTED FUZZY ENERGY & SECURITY AWARE SCHEDULING IN CLOUD

[1]SIRISATI RANGA SWAMY, [2]Dr SRIDHAR MANDAPATI
[1]Research Scholar, Acharya Nagarjuna University, Department of Computer Science, Guntur, Andhra Pradesh, India
[2]Associate Professor. RVR&JC College of Engineering, Department of Computer Application, Guntur, Andhra Pradesh, India
E-mail: [1]sirisatiranga@gmail.com, [2]mandapati_s@yahoo.com

ABSTRACT

The paradigm in which Information Technology (IT) application is provided as a service is referred to as cloud computing. It enables the users to make commercialized use of computation, storage, data and services across the world. The security of this model is questionable despite the fact that several potential gains can be achieved through cloud computing and this deters the adoption of the cloud computing model. One of the main tasks that is performed in cloud computing is Job scheduling. This not only improvises the working of the cloud environments but also enables achieving maximum profit. Due to its large solution space and the large amount of time required to find an optimal solution, Scheduling is a part of problem category known as Non-deterministic Polynomial (NP)-hard problem. This work proposes the usage of fuzzy logic as well as optimized rule selection for cloud scheduling. A fuzzy logic system that maps the input space into an output space using a list of if-then statements referred to as rules is known as the fuzzy inference system. Here Stochastic Diffusion Search (SDS) algorithm based fuzzy rule selection is proposed. The SDS as the basis of the approach and modified it with the aid of fuzzy theory to assign jobs to the most suitable resources. It has been shown that the suggested technique helps to achieve better performance than fuzzy logic, as per experimental outcomes.

Keywords: *Cloud Computing, Job Scheduling, Cloud Security, Fuzzy Logic, Rule Selection and Stochastic Diffusion Search (SDS)*

1. INTRODUCTION

In cloud computing paradigm, computer is used more as a service than a product by rendering utility in the form of services that can be shared including the data and the software to end users over internet. Going by the cloud users and the service providers, the reduction in the capital expenditures of the systems is one prime advantage of cloud computing. For a wide range of such applications including the hosting websites, scientific workflows, customer services, e-business, high performance computing and social networking in the paradigm of cloud computing has been found to be useful [1].

Cloud computing provides viable models which are- cloud Infrastructure as a Service (IaaS), cloud Platform as Service (PaaS) and cloud Software as a Service (SaaS). In Cloud IaaS, a consumer can process, store and exploit storage and networks by running arbitrary software that comprise operating systems and applications as well. In cloud PaaS, in order to develop an application, developers are facilitated with provider specific programming language and tools. Users can use the provider's application running on cloud infrastructures using the Cloud SaaS. Through client interfaces such as the web browsers, the applications can be accessed from various client devices [2].

The various types of cloud computing are as follows: Public clouds- Here, the services may be used by anyone who registers. The Private Clouds- Here, with no restrictions of network bandwidth, the legal requirements and the security exposures along with the data processes that have been managed inside that of the organizations. The Hybrid Clouds bright the private and the extrinsic clouds together. The virtualization is that enabling technology used in cloud computing which will be able to segregate one single machine into many of the Virtual machines (VMs) will be made in a cost effective manner. The VM permits the platform for being tailored according to the needs of that of the end user and also for the isolation of such applications from all the underlying hardware or also the other VMs. Such facilities have been presented by the Cloud Service provider (CPS) that can pay per use [3].

In case of cloud computing the location independent infrastructure that is of low cost and scalable has been provided for the storage as well as management of data. Such increasing data volumes have been stored at the remote servers and also accompanies the rapid adoption of the services of cloud and therefore there will not be any need for the techniques of saving the disk space and its network bandwidth. Irrespective of how many clients actually request a file, the when only one single copy of every file is stored by the server, the concept is de-duplication that is gaining a lot of attention. All the clients storing the file just make use of the links to only one copy of the file that is stored at the server .Also, the clients need not upload the file again in case the server already has the file copy and thus, not only the bandwidth but also the storage is saved and this is referred to as client side de-duplication. Which again introduced new security issues [4].

There are factors other than authenticity, authorization and responsibility, such as data protection, disaster recovery and business continuity that are related to security in cloud computing. As cloud computing involves giving up direct control over several aspects of security and privacy, this very nature of cloud computing makes its security a complex task. This is why, several enterprises hesitate to co-host their internal company data on external servers that may need to be shared with other enterprises. Also, it is imperative for the organizations to have a high degree of trust on their cloud service providers as the service providers may have access to all their data which they may accidentally or deliberately misuse [5].

In Scheduling, the available resources are assigned on the basis of the needs and the nature of the tasks seeking them. Increased utilization of the resources without impacting the services that are provided by the cloud is the main goal of scheduling. This task is challenging as available resources are to be allocated on the basis of the qualities, metrics and the requirements of the tasks. Scheduling decides the manner in which crucial resources like I/O, network, memory, CPU, secondary storage space and network bandwidth are allocated between the users and their tasks. An efficient scheduling strategy accounts for evolving environments, nature of the tasks and the task metrics when a batch is submitted for processing [6].

A basic concept of cloud computing systems is Job management, in addition to task scheduling which is responsible for the efficient working of the cloud computing system. Job scheduling maps the tasks to the appropriate resources and it is flexible and convenient. Based on the priorities, needs and functions of a business, jobs as well as jobs streams can be scheduled to run . These job streams can be scheduled to run daily/weekly/monthly/yearly without the requirement of any support staff [7].

The desired Quality of Service (QoS) can be easily attainable by means of scheduling which allocates the resources among the given tasks in a finite time. Technically speaking, in scheduling, based on resource restrictions, tasks are scheduled and some objective function is optimized. This problem of mapping tasks to a seemingly infinite resource pool in cloud computing belongs to a problem category referred to as NP hard problems. For such problems, there are no algorithms that can provide an optimal solution within the polynomial time. The operating expenses of generating schedules are extremely high and hence, solutions that are based on exhaustive search are infeasible. Meta heuristic techniques however, can provide a near optimal solution in a reasonable timeframe. As they are both effective as well as efficient in solving large and complex problems, meta heuristics have become largely popular [8].

In several fields of life ranging including medicine, fuzzy logic is a solution as it is very similar to human reasoning as well as decision making. Where everything cannot be described in accurate and discrete terms, fuzzy logic looks into all the shades of gray and also answers the various ambiguities and uncertainties that are created by human language. This system can help us understand to what extent the disease has progressed and it can also account for the risks and the other morbidities for an individual patient. However, there are several disadvantages as well- Developing fuzzy rules and membership functions is a complicated task and as the fuzzy rules can be interpreted in different ways, its analysis is extremely challenging. Also, there is no standard result and it is necessary to run the program for each individual patient .Hence, in the absence of pre-programmed software for various pathologies, and the basic training for the clinicians for using this software, its clinical applicability and utilization poses difficulties [9].

In this work, the optimal rules (i.e. fuzzy rules) are selected using SDS is proposed for cloud scheduling. The following sections comprise the remainder of the investigation -. Section 2 discusses related works in literature. Section 3 explains various methods used in the work. Section 4 discusses experimental results and Section 5 concludes the work.

2. RELATED WORKS

An optimal integrity policy for enhanced data security and integrity in the cloud was proposed by Kumari & Kamal [10]. In secret key generation, this work use AND, EXOR and hashing operations to improve the security. The effectiveness of encryption time over existing Ciphertext Policy-Attribute Based Cryptography (CP-ABE) methodologies is confirmed by an optimal integrity policy based on the performance measure. Better performance is shown in Office of Infrastructure Protection (OIP) than the Efficient Privacy-Preserving Demand Response (EPPDR), The subset cover with a pseudo random key generation makes use of a comparative analysis of all parameters including the computational overheads, the average time for its generation and the key derivation. The data security has also been enhanced by the increase in its secret key generation.

The intelligent approach to cryptography that is the cloud service operation that will not directly be able to reach the partial data and had been suggested by Li et al [11]. According to the suggested approach, the file is divided and the data is separately stored in distributed cloud servers. Alternate technique was formulated for finding if the data packets had to be split so that the time of operation could have been shortened. Mainly supported by the proposed algorithms, including Alternative Data Distribution (AD2) Algorithm, Secure Efficient Data Distributions (SED2) Algorithm and Efficient Data Conflation (EDCon) Algorithm, the proposed scheme is entitled Security-Aware Efficient Distributed Storage (SA-EDS) model, which is.

Rimal & Maier [12] came up with a new policy which used idle resources in a better way and also reduced the overall workflow finish time and the cost of workflows as well. This was a novel Cloud-Based Workflow Scheduling Algorithm (CWSA) policy for computer intensive workflow applications in multi-tenant cloud computing environments. For performance evaluation, the suggested algorithm has been compared with the other algorithms like First Come First Served (FCFS), EASY Backfilling, and Minimum Completion Time (MCT) scheduling policies. Furthermore, for demonstrating the scalability of the CWSA that verifies the efficacy of the suggested solution, a proof-of-concept experiment of the real world scientific workflow applications has been performed.

Fuzzy logic was adopted by Amini et al [13] for dealing with issues like insufficient data as well as estimating the severity and the probability of every risk mathematically. Developing a notional model for prioritizing risks on the basis of probability and severity is the objective of this work. It is essential to integrate human knowledge and expertise into a role based circumstance for estimating risk. Hence, this work presents fuzzy logic and for converting linguistics data to numerical value so that the risk rate is quantified. However, as insufficient information is available for obtaining the quantitative data due to risk characterizing factors, fuzzy logic has been used to deal with human experiences.

The effective resource management in cloud computing environment is contributed by Migration. Hence in several areas such as load balancing in cloud computing Data Centers (CDCs), migration has been used. But, the minimization of migration time has been the area of concentration of most of the previous research work. Also, generally balance between the QoS metrics is not considered by the prior works. Hence, combining multiple metrics has to be considered due to the importance of balance of metrics. QoS metrics of migration was presented by . Son & Huh [14], who suggested a migration technique that depended on fuzzy logic. Applying fuzzy logic and machine learning technique for advanced migration is the main objective of this work.

Arabnejad et al., [15] compared two dynamic learning strategies based on a fuzzy logic system, which learns and modifies fuzzy scaling rules at runtime. A self-adaptive fuzzy logic controller is combined with two Reinforcement Learning (RL) approaches: (i) Fuzzy State Action Reward State Action (SARSA) Learning (FSL) and (ii) Fuzzy Q-Learning (FQL). As an off-policy approach, Q-learning learns independent of the policy currently followed, whereas SARSA as an on-policy always incorporates the actual agent's

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

behavior and leads to faster learning. Both approaches are implemented and compared in their advantages and disadvantages, here in the OpenStack cloud platform. This work demonstrated that both auto-scaling approaches could manage varying load traffic, sudden and periodic, and while reducing operating costs and preventing Service Level Agreement (SLA) violations, managed to deliver resources on demand. The experimental results demonstrate that FSL and FQL have acceptable performance in terms of adjusted number of VM targeted to optimize SLA compliance and response time.

For decreasing the consumption of power as well as the processing time and also for improving the service provider's profits, through decreased operational expenses and improved system reliability, task scheduling is an extremely important task. For optimizing the makespan and waiting time, Alla et al.,[16] focussed on task scheduling by making use of a new type of architecture with dynamic queues that were based on hybrid algorithm that made use of fuzzy logic and Particle Swarm Optimization (TSDQ-FLPSO). It was shown by experimental result based on open source simulator (CloudSim) that by this, optimal balanced results were obtained which not only reduced the waiting time but also improvised the utilization of resources when compared to the existing scheduling algorithms.

To solve the issue of computing resources in a cloud manufacturing system, Lin & Chong [17] incorporated several novel ideas through local search and enhancements and presented a Genetic Algorithm (GA) based resource constraint project scheduling. As the task took priority, and the availability of resource constraints was present, a newly generated offspring may be infeasible. For finding better schedules, the neighbourhood of the solutions can be exploited by the local search. The allocation of computing resources in cloud manufacturing system is NP-hard because of its complex characteristics.

Based on the Firefly Algorithm (FA), Kansal& Chana [18] proposed an energy-aware VM migration technique for cloud computing. .The suggested method maintains the performance as well as the energy efficiency of the data centres by migrating the maximally loaded VMs to the least loaded active node .Using the CloudSim simulator, the efficacy of the suggested technique has been shown by comparing it with the other techniques. A decrease in a mean of 72.34 % of migrations and salvaging 34.36 % of hosts, the average consumption of energy went up by 44.39 %. Thus , data centers are better at conserving power.

To provision resources and schedule workflows in IaaS environment, Kaur & Mehta [19] presented an Augmented Shuffled Frog Leaping Algorithm (ASFLA) based approach. This was weighed against the PSO and the SFLA and found that when custom based Java simulator was used some popular scientific workflows of different sizes assessed the efficiency of ASFLA .In terms of attaining least execution costs and meeting scheduling deadlines, there was a marked improvement shown by simulation results.

For handling the workflow scheduling problem with multiple conflicting objective functions on IaaS clouds, Verma & Kaushal [20] presented a non-dominance sort based Hybrid Particle Swarm Optimization (HPSO) algorithm. The proposed algorithm is a hybrid of the previously proposed Budget and Deadline constrained Heterogeneous Earliest Finish Time (BDHEFT) algorithm and multi-objective PSO. Two conflicting objectives-makespan and cost are attempted to be optimized by the HPSO; the constraints are deadline and budget .Besides the conflicting objectives, there is also a reduction of the power that has been consumed by the workflow that is created. A set of Pareto optimal solutions result from the suggested algorithm from wherein the user can select the best solution. The solutions that are obtained, substantiated by the simulation analysis, with the suggested algorithm deliver a better convergence as well as uniform spacing compared to the other heuristics.

3. METHODOLOGY

In cloud computing, maximal resource utilization assures that resources are allocated to jobs with optimal time and cost by an algorithm. Taking into consideration the user preferences and requirements, assigning the tasks to the most viable resources is the most critical issue for job scheduling. This section discussed using SDS algorithm as well as fuzzy logic.

3.1 Fuzzy Logic

When there is vague/noisy/missing input information, a definite conclusion can be arrived at by using the fuzzy logic. The type of logic matters more than the simple true or false values. Zadeh, proposed the fuzzy logic in 1965; as it has the ability to deal with imprecise as well as incorrect information, it is now used widely. The human expertise is embedded in the system as the fuzzy logic is very close to the human mind. In various fields such as washing machines, image processing, microprocessors, air conditioners and microcontrollers, fuzzy logic has been widely used.

Having the ability of handling the inaccurate inputs there is a fuzzy inference that has been used widely for the solving of the control along with the problems of reasoning in the environments that are uncertain. There are also three other major components for a typical inference model [21]:

The Inference Engine: this defines all fuzzy logic operators with their defuzzifier that has been used in the inference process.

The Membership Functions: this degree of the fuzzy element will belong to its corresponding fuzzy set that has been defined in the membership function. The mapping of the crisp values to their degrees of membership which can vary between that of 0 and 1 and each input and output variable will have a similar set of the membership functions.

Rule base: The inference model that is defined by the "If-Then" rule comprises this set. The "AND" or "OR" operators connect the antecedent and consequent fuzzy propositions in "If antecedent Then consequent". This is how a rule structure looks like.

There are 5 main steps in the inference procedure:

1) Fuzzification: For obtaining the corresponding membership degrees of every input variable, with regard to particular fuzzy set, input crisp values into membership functions.

2) The applying of Fuzzy Operations: this is for obtaining the membership of degree for the antecedent using the "AND" and the "OR" operators.

3) Implication: Use the defined implication operator for obtaining the fuzzy set of each and every rule.

4) Aggregation: an aggregate of output fuzzy sets for all the rules that make use of a defined aggregation operator.

5) Defuzzification: using a defined defuzzification algorithm that is for transforming all the collected fuzzy set within a crisp value.

The inference of fuzzy logic flow from an input variable to all the output variables that have been identified by this system structure. The analog inputs have been translated into that of fuzzy values using the fuzzification in their input interfaces. These rule blocks a compromise of the rules of linguistic control in which a fuzzy inference will take place and the linguistic variables will be the outputs of this particular rule block. They get translated into the different analog variables by a defuzzification within the interfaces of the output and the entire structure of this fuzzy system that includes the interfaces of the input, the rule blocks and the output interfaces has been shown in figure 1. The data flow is shown in the connecting lines.

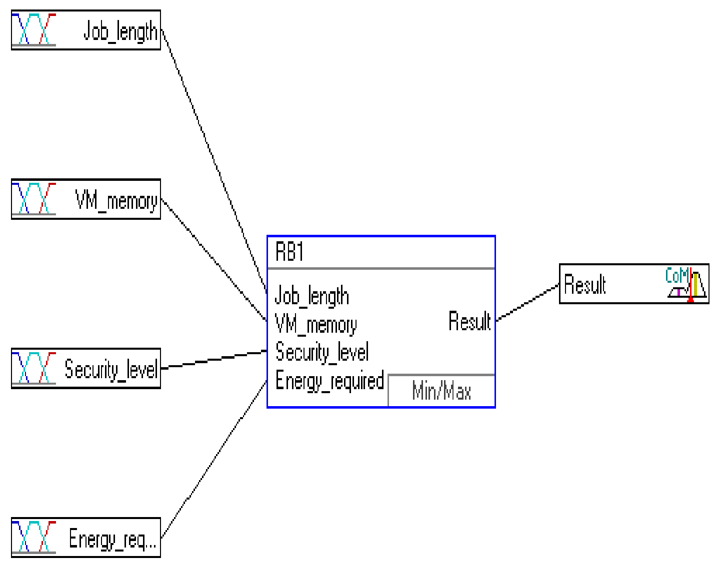


Figure 1: Structure of the Fuzzy Logic System

If no rule firing for this variable, the default value of an output variable is used. The resultant for different methods of defuzzification can be either ‚most plausible result' or ‚best compromise'. The latter is produced by the techniques:Center of Maximum (CoM), Center of Area (CoA) and CoA BSUM, a version especially for efficient VLSI implementations. The 'most plausible result is produced by the methods: MoM (Mean of Maximum (MoM) and MoM BSUM, a version especially for efficient VLSI implementations.

The MBF of energy required, job length, security level, VM memory and result as shown in figures 2, 3, 4, 5, 6.

© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

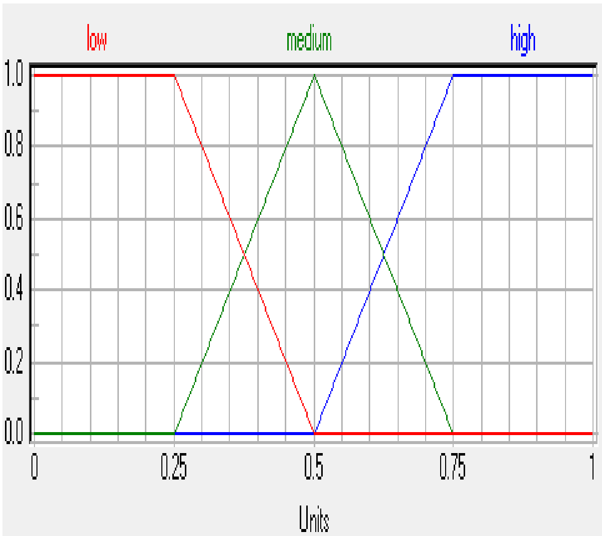


Figure 2: MBF of "Energy_required"

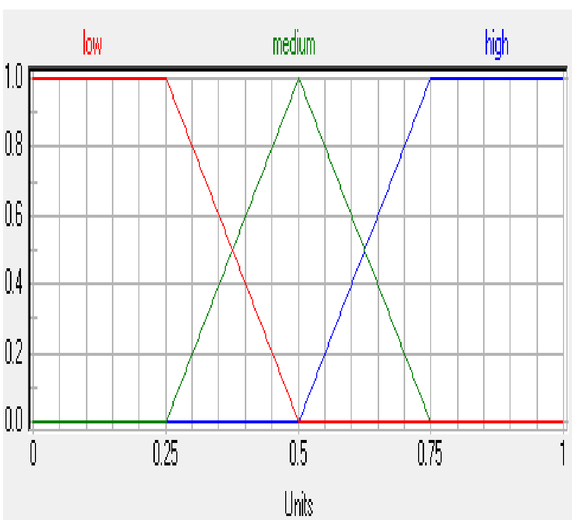
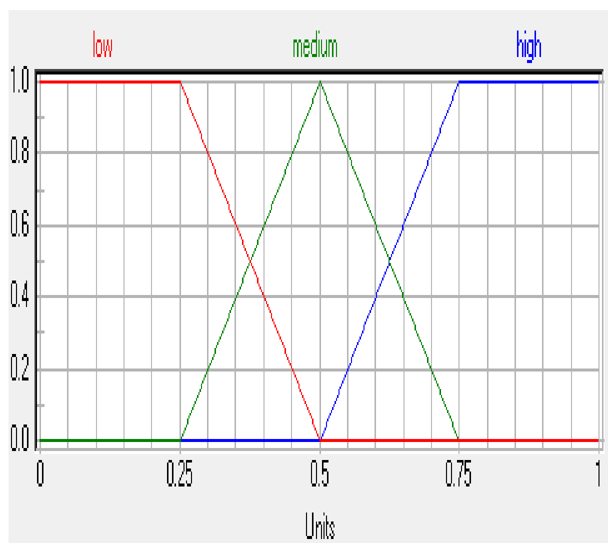
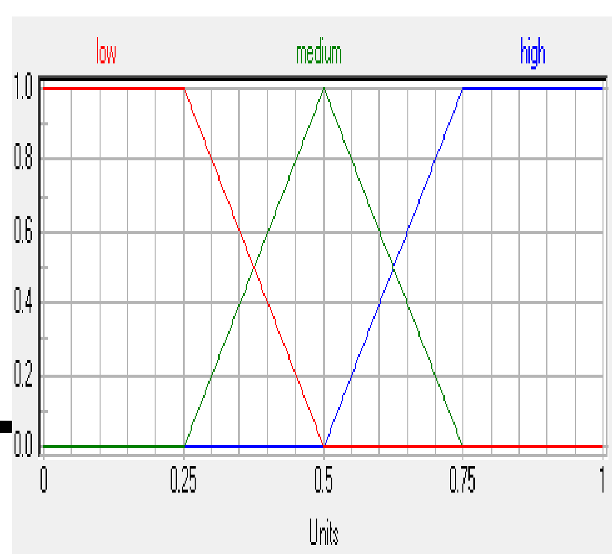


Figure 3: MBF of "Job_length"

Figure 4: MBF of "Security_level"

Figure 5: MBF of "VM_memory"





Journal of Theoretical and Applied Information Technology
31st May 2018. Vol.96. No 10
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

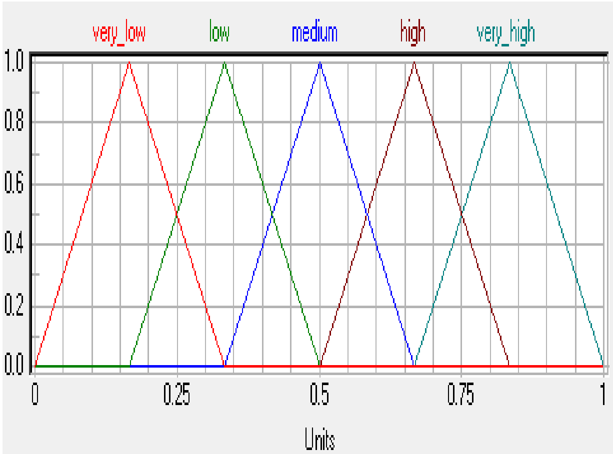


Figure 6: MBF of "Result"

A control strategy for the system of fuzzy logic has been included in their rule blocks and for the similar context all of the rules will be confined by each rule block. A similar input as well as an outpur variable for the rules will define this context and in this the fuzzy system response will be described in the then path and the situation in which the rules have been designed in the if part. Every rule has been weighted in accordance to the importance of its Degree of Support (DoS). Calculating the if part will be the beginning of the rules of processing and this method has been used for the determination of the type of operator of this rule. The operator will also type the MIN-MAX, the MIN-AVG and the GAMMA that are now available and the characteristic of every type of operator will be influenced using an additional parameter.

For example:

MIN-MAX,
parameter value 0=Minimum Operator (MIN)

MIN-MAX,
parameter value 1=Maximum Operator (MAX)

GAMMA,
parameter value 0=Product Operator(PROD)

A minimum operator will be a generalization of a Boolean that is "and" and its maximum operator will be a generalization of a Boolean "or" where various rules have been combined eventually into a single conclusion by its fuzzy composition. All the firings have been evaluated using the BSUM method and has been used with the dominant rules that have been evaluated in the max method that is used.

3.2 Proposed Rule Selection using SDS

It is possible to consider fuzzy control rule as the knowledge of an virtuoso in any field. A sequence of the form IF-THEN represent the fuzzy rule; this in turn leads to algorithms that can describe the action or the output which is to be taken with regard to the currently observed information: the one that may comprise input as well as the feedback in case a closed-loop control system is applied. The experiences and knowledge base of the humans forms the basis of the law for designing a set of fuzzy rules [22].

A condition is associated by a fuzzy IF-THEN rule which is described utilizing fuzzy sets as well as linguistic variables to an output or conclusion. Knowledge is captured using elastic conditions by the IF part; Conclusion or output can be given in linguistic variable form by the THEN part. For the purpose of evaluation of the degree of input matching of data with a condition of the rule takes place IF-THEN rule has been used widely by the fuzzy inference system.

The fuzzy mapping rules will provide one more functional mapping among the input as well as the output variables by using its linguistic variables. In the real life situations, it can also be a challenge for deriving the relationship among the input as well as the output and there may also be a correlation that is formulated. In such situations, the fuzzy mapping rules will be one good solution [23].

The manner in which the functions of fuzzy mapping will be quite similar to its human insight. For the whole function to be approximated using a fuzzy mapping rules, every rule of fuzzy mapping will approximate one restricted set of elements of a function. By still making use of the air conditioner system to be its example the rule of fuzzy mapping is derived as IF its temperature is found to be LOW, THEN a heater motor will have to be rotated FAST.

The rule of fuzzy implication is a generic logic based implication and its correlation among the inputs and the outputs. This has a two-valued logic and also a multiple valued logic. By still making use of the system of air conditioner as its example, an implication will be IF and the temperature being LOW, THEN its heater motor will have to be FAST. Based on this the temperature will be found to be HIGH. The result

Journal of Theoretical and Applied Information Technology
31st May 2018. Vol.96. No 10
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

will be that the heater motor will slow down or a SLOW may be inferred.

This SDS, being a multi-agent global search along with an optimization algorithm will have a simple interaction of the agents and there are some resources assigned using the processes demonstrated with a high level description shown as a social metaphor. This SDS will further introduce one more probabilistic approach which will be able to tackle a pattern recognition that is best fit and can also match the problems. This makes use of the interaction among the simple agents and also the multi agent population that has been based on the global search with it being a distributive technique for computing. Aside from this convergence being global optimum, the criteria for minimal convergence with a linear time complexity, a robust framework of mathematics has been associated using the SDS for the behavior of the algorithm by means of looking into the allocation of resources [24].

The algorithm of SDS will commence the search or also its optimization by means of initializing the population (for example its miners and the metaphor of the mining game). For any of the SDS searches every agent will maintain a hypothesis h that defines the possible solution and for this mining game analogy, the agent hypothesis will identify the hill.

The algorithm for SDS is as below:

```
01: Initialising agents ()
02: While (stopping condition is not met)
03: Testing hypotheses ()
04: Diffusion hypotheses ()
05: End While
```

All the agents will randomly select the hypothesis from its search space at the time of initialization. The agents are set to be inactive and access is given to its target model. Random initializations for these positions are biased in favor of certain positions that are described in the a-priori knowledge. There have been two different types of knowledge that can influence this initialization phase [25]:

(i) The actual ratio of the model's size to the search space and its size will be greater than that of one and this will ensure that the agent is well initialized using the best of hypothesis.

(ii) The earlier location of this model is known and can be used in performing the other searchers for similar spaces like frames of video.

If the agent hypothesis is fruitful then this will not be checked within the test phase by means of conducting an evaluation of partial hypothesis which will return a Boolean value. Based on an accurate approach to recruitment which is used in its iteration there is a successful hypotheses that will diffuse across its population and this is the way in which the solutions that is potentially viable will be disseminated in the agent population [26].

In case of the test phase, every agent will perform one partial function evaluation, the pFE, that is a function of the hypothesis of the agent; the pFE = f (h). Here in the mining game a partial function evaluation will entail the mining of a random selected region on that of the hill that has been defined by the hypothesis of the agent.

For the interaction and the communication of its hypothesis each agent in its diffusion phase will recruit one more and the diffusion will be performed by means of communicating the hill hypothesis in the metaphor of mining game.

This fuzzy approach will make use of this for choosing the best of the two SDS phase that is as below: there are some fuzzy rules that have been fired on the basis of the input parameters. In a manner in which the decisions have been made will be for choosing an ideal SDS and this is made according to the aggregation of its fired fuzzy rules, that are integrated. These fuzzy sets will show such outputs of its fired fuzzy rules integrated into one single fuzzy set. The output of its defuzzification will be a non-fizzy number. There are also five other common defuzzification based methods that are: centroid, mean of maximum, largest value of maximum, smallest value of maximum and bisector [27].

The area's center of gravity is under a curve that has been calculated using this centroid method and there is a vertical line that divides the region for two of the equal sub-regions of the area that has been carried using the bisector method. The mean of the maximum approach will involve a plateau that is at its peak value for the final function that takes the mean of the value which is on the covers. At the same time, the smallest of the values of its maximum approach that will involve the peak value of its final function will assume the values that are covered and has a greatest value. This centroid method will favor the largest value of its maximum. One of the most commonly used method

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

is this centroid method. As per Equation (1) the centroid method is presented in which the μ shows the degree of membership of fired rules.

$$a = \frac{\int_z \mu_A(z) z dz}{\int_z \mu_A(z) dz} \quad (1)$$

In this definition, a will be the outcome of a rule of this method of a percentage of the risk that has been associated with this. The fitness value of this SDS hypothesis will make use of a non-fuzzy number. The fuzzy function will compute its fitness value and the fitness values that are computed for every such job gives the fitness function of all the hypothesis. The most ideal of the two SDS phases for each type that has been chosen that is based on the fitness values that are calculated, an ideal SDS phase of the very first type with the best SDS phase of that of the second-type.

There is also an exchange of resources that was assigned to its SDS agents (the very first agent of the first of the hypothesis of the first agent of its second hypothesis and the second of the first hypothesis having a second agent of its second hypothesis) that is sued in the fuzzy system output. For the purpose of reduction of the make span and its cost the agents of this hypothesis will also assign some suitable resources to their jobs.

The operation will further continue till such time the output hypothesis of both the types have been in the homolog meaning which have both and the exactly similar hypothesis. The whole process is now repeated using the recursive algorithm and the breaking condition will be the number of iteration or by the chosen hypothesis that is the same (the same VMs positions and the jobs).

4. RESULTS AND DISCUSSION

In this section, the fuzzy and rule selected fuzzy methods are used. The parameter setting as shown in table 1. The makespan, degree of imbalance and resource utilization as shown in tables 2, 3, 4 and figures 7, 8, 9.

Table 1: Parameter Setting

Parameters	Values
Length of job	1,000–20,000
Total number of jobs	100–1,000
Total number of VMs	50
VM frequency	500–2,000
Population size	10
VM memory (RAM)	256–2,048
VM bandwidth	500–1,000
Number of PEs requirements	1-4
Number of datacenters	10
Number of hosts	2-6

Table 2: Makespan for Rule Selected Fuzzy

Number of Jobs	Fuzzy	Rule Selected Fuzzy
200	44	40
400	92	86
600	147	138
800	193	182
1000	247	226

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

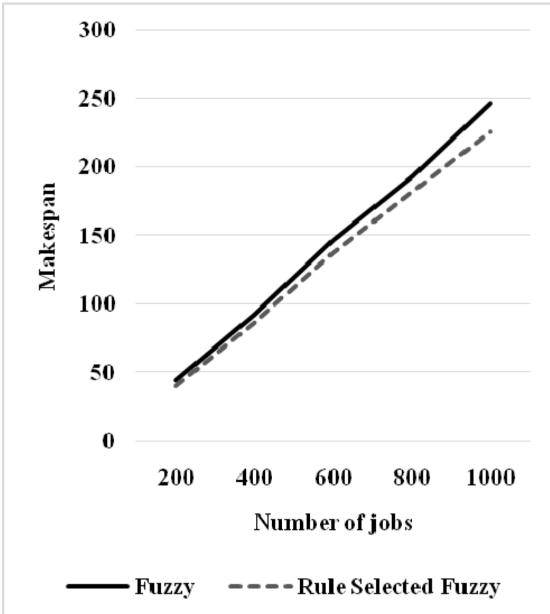


Figure 7: Makespan for Rule Selected Fuzzy

From the figure 7, it can be observed that the rule selected fuzzy has lower makespan by 9.52% for 200 number of jobs, by 6.74% for 400 number of jobs, by 6.31% for 600 number of jobs, by 5.86% for 800 number of jobs and by 8.87% for 1000 number of jobs when compared with fuzzy.

Table 3: Degree of Imbalance for Rule Selected Fuzzy

Number of Jobs	Fuzzy	Rule Selected Fuzzy
200	2.5	2.3
400	2.4	2.3
600	2.1	2
800	2	1.9
1000	1.9	1.7

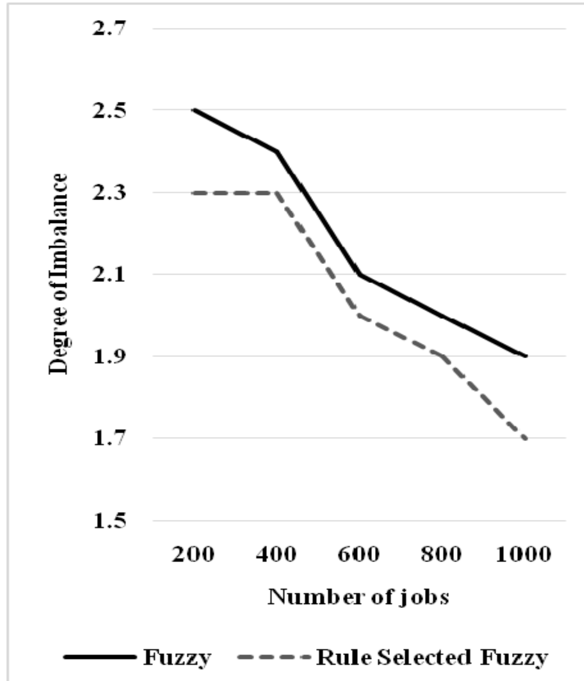


Figure 8: Degree of Imbalance for Rule Selected Fuzzy

From the figure 8, it can be observed that the rule selected fuzzy has lower degree of imbalance by 8.33% for 200 number of jobs, by 4.25% for 400 number of jobs, by 4.87% for 600 number of jobs, by 5.12% for 800 number of jobs and by 11.11% for 1000 number of jobs when compared with fuzzy.

Table 4: Resource Utilization for Rule Selected Fuzzy

Number of jobs	Fuzzy	Rule Selected Fuzzy
200	79	82
400	78	81
600	81	84
800	82	85
1000	79	81

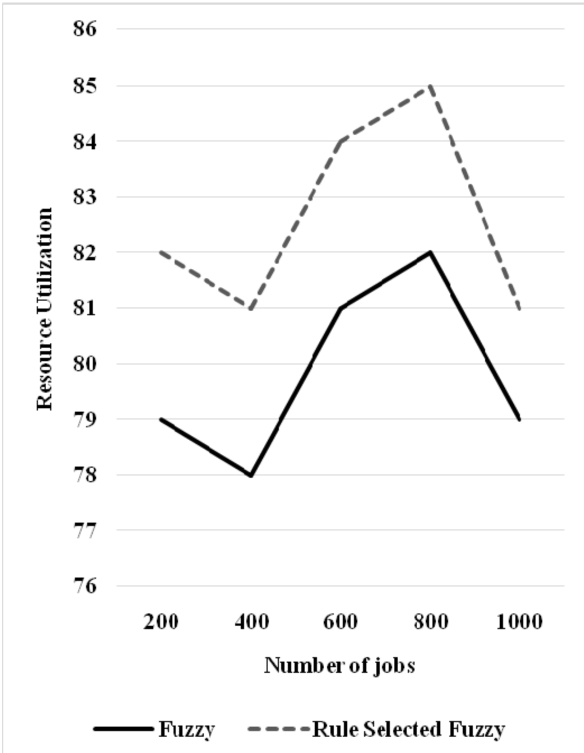


Figure 9: Resource Utilization for Rule Selected Fuzzy

From the figure 9, it can be observed that the rule selected fuzzy has higher resource utilization by 3.72% for 200 number of jobs, by 3.77% for 400 number of jobs, by 3.63% for 600 number of jobs, by 3.59% for 800 number of jobs and by 2.5% for 1000 number of jobs when compared with fuzzy.

5. CONCLUSION

There are several technological applications across the internet for the cloud computing technologies. There has been a paradigm shift in the IT world with the advent of cloud computing. There has been a maximization of profits and with the newer applications being formulated, the complexity of the scheduling process has increased proportionally. Thus, for ensuring that there is a maximum return of investment on the resources (ROI), job scheduling plays a very important role so that the resources are effectively utilized. The work concentrated on the research areas of job scheduling in cloud computing. For the input linguistic variables that constitute the premise, the fuzzy rule-base is usually generated as an exhaustive set of all possible value- combinations. It presents a high level description of SDS in the form of a social metaphor that shows the procedures for assigning the resources for SDS. For solving the best-fit pattern recognition and matching problems ,A novel probabilistic approach is introduced by the SDS .The SDS hypothesis is created by : The proposed approach in which jobs are represented as agents; and computational resources assigned to these agents, and sets of agents. Results show that the rule selected fuzzy has higher resource utilization by 3.72% for 200 number of jobs, by 3.77% for 400 number of jobs, by 3.63% for 600 number of jobs, by 3.59% for 800 number of jobs and by 2.5% for 1000 number of jobs when compared with fuzzy.

REFRENCES:

[1] Panda, S. K., & Jana, P. K. (2015). Efficient task scheduling algorithms for heterogeneous multi-cloud environment. The Journal of Supercomputing, 71(4), 1505-1533.

[2] Kaur, S., & Verma, A. (2012). An efficient approach to genetic algorithm for task scheduling in cloud computing environment. International Journal of Information Technology and Computer Science (IJITCS), 4(10), 74.

[3] Masdari, M., Salehi, F., Jalali, M., & Bidaki, M. (2017). A Survey of PSO-based scheduling algorithms in cloud computing. Journal of Network and Systems Management, 25(1), 122-158.

[4] Canedo, E. D., de Sousa Junior, R. T., & de Oliveira Albuquerque, R. (2012). Trust model for reliable file exchange in cloud computing. International Journal of Computer Science & Information Technology, 4(1), 1.

[5] Benabied, S., Zitouni, A., & Djoudi, M. (2015, June). A cloud security framework based on trust model and mobile agent. In Cloud Technologies and Applications (CloudTech), 2015 International Conference on (pp. 1-8). IEEE.

[6] Varma, P. S., & Krishna, S. R. (2016). Design and Development of Efficient Cloud Scheduling Algorithm based on Load Balancing Analytics. International Journal of Computer Applications, 138(12).

[7] Kaur, R., & Kinger, S. (2014). Analysis of job scheduling algorithms in cloud computing. International Journal of Computer Trends and Technology (IJCTT), 9(7), 379-386.

[8] Kalra, M., & Singh, S. (2015). A review of metaheuristic scheduling techniques in cloud

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

computing. Egyptian informatics journal, 16(3), 275-295.

[9] Godil, S., Shamim, M., Enam, S., & Qidwai, U. (2011). Fuzzy logic: A" simple" solution for complexities in neurosciences?. Surgical neurology international, 2, 24.

[10] Kumari, P. S., & Kamal, A. N. B. (2016). Optimal Integrity policy for encrypted data in secure storage using cloud computing. Indian Journal of Science and Technology, 9(8).

[11] Li, Y., Gai, K., Qiu, L., Qiu, M., & Zhao, H. (2017). Intelligent cryptography approach for secure distributed big data storage in cloud computing. Information Sciences, 387, 103-115.

[12] Rimal, B. P., & Maier, M. (2017). Workflow scheduling in multi-tenant cloud computing environments. IEEE Transactions on Parallel and Distributed Systems, 28(1), 290-304.

[13] Amini, A., Jamil, N., Ahmad, A. R., & Sulaiman, H. (2017, April). A Fuzzy Logic Based Risk Assessment Approach for Evaluating and Prioritizing Risks in Cloud Computing Environment. In International Conference of Reliable Information and Communication Technology (pp. 650-659). Springer, Cham.

[14] Son, A. Y., & Huh, E. N. (2017, July). Study on a migration scheme by fuzzy-logic-based learning and decision approach for QoS in cloud computing. In Ubiquitous and Future Networks (ICUFN), 2017 Ninth International Conference on (pp. 507-512). IEEE.

[15] Arabnejad, H., Pahl, C., Jamshidi, P., & Estrada, G. (2017, May). A Comparison of Reinforcement Learning Techniques for Fuzzy Cloud Auto-Scaling. In Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (pp. 64-73). IEEE Press.

[16] Alla, H. B., Alla, S. B., Ezzati, A., & Mouhsen, A. (2017). A novel architecture with dynamic queues based on fuzzy logic and particle swarm optimization algorithm for task scheduling in cloud computing. In Advances in Ubiquitous Networking 2 (pp. 205-217). Springer Singapore.

[17] Lin, Y. K., & Chong, C. S. (2017). Fast GA-based project scheduling for computing resources allocation in a cloud manufacturing system. Journal of Intelligent Manufacturing, 28(5), 1189-1201.

[18] Kansal, N. J., & Chana, I. (2016). Energy-aware Virtual Machine Migration for Cloud Computing-A Firefly Optimization Approach. Journal of Grid Computing, 14(2), 327-345.

[19] Kaur, P., & Mehta, S. (2017). Resource provisioning and work flow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm. Journal of Parallel and Distributed Computing, 101, 41-50.

[20] Verma, A., & Kaushal, S. (2017). A hybrid multi-objective Particle Swarm Optimization for scientific workflow scheduling. Parallel Computing, 62, 1-19.

[21] Qu, C., & Buyya, R. (2014, May). A cloud trust evaluation system using hierarchical fuzzy inference system for service selection. In Advanced information networking and applications (aina), 2014 ieee 28th international conference on (pp. 850-857). IEEE.

[22] Prasath, V., Bharathan, N., Lakshmi, N., & Nathiya, M. (2013). Fuzzy logic in cloud computing. Int. J. Eng. Res. Technol, 2(3), 1-5.

[23] Bai, Y., & Wang, D. (2006). Fundamentals of fuzzy logic control—fuzzy sets, fuzzy rules and defuzzifications. In Advanced Fuzzy Logic Technologies in Industrial Applications (pp. 17-36). Springer London.

[24] El-henawy, I. M., & Ismail, M. M. (2014). A Hybrid Swarm Intelligence Technique for Solving Integer Multi-objective Problems. International Journal of Computer Applications, 87(3).

[25] Williams, H., & Bishop, M. (2014). Stochastic diffusion search: a comparison of swarm intelligence parameter estimation algorithms with ransac. Algorithms, 7(2), 206-228.

[26] al-Rifaie, M. M., Bishop, J. M., & Blackwell, T. (2012). Information sharing impact of stochastic diffusion search on differential evolution algorithm. Memetic Computing, 4(4), 327-338.

[27] Shojafar, M., Javanmardi, S., Abolfazli, S., & Cordeschi, N. (2015). FUGE: A joint meta-heuristic approach to cloud job scheduling algorithm using fuzzy theory and a genetic method. Cluster Computing, 18(2), 829-844.